

# Min-Sum algorithm for lattices constructed by Construction $\mathbf{D}$

Hassan Mehri

*Department of Mathematics and Computer Science, Amirkabir University of Technology,  
Tehran, Iran*

---

## Abstract

The so-called min-sum algorithm has been applied for decoding lattices constructed by Construction  $\mathbf{D}'$ . We generalize this iterative decoding algorithm to decode lattices constructed by Construction  $\mathbf{D}$ . An upper bound on the decoding complexity per iteration, in terms of coding gain, label group sizes of the lattice and other factors is derived. We show that iterative decoding of LDGM lattices has a reasonably low complexity such that lattices with dimensions of a few thousands can be easily decoded.

*Keywords:* Lattices, Iterative decoding, Min-Sum algorithm, LDGM codes

---

## 1. Introduction

Both of the integer programming method and the trellis approach, as two main methods for lattice decoding, are impractical in higher dimensions [4, 5]. The min-sum algorithm, as an iterative decoding approach, can be used in decoding high dimensional lattices. Tanner generalized Gallager's low-density parity-check (LDPC) codes to other class of codes defined by

---

*Email address:* hassanmehri.math@gmail.com (Hassan Mehri)

general bipartite graphs, called Tanner graphs [7]. For any linear binary block code, this construction is based on a parity check matrix of the code. Tanner graph construction is used to find a graphical representation of the lattice in terms of its linear constraints. The decoding complexity of the generalized min-sum algorithm depends on the Tanner graph structure and the label code of the lattice. Sadeghi et al. introduced a generalization of min-sum decoding algorithm for lattices constructed by Construction  $\mathbf{D}'$  [6]. In this work, we will propose another generalization of min-sum algorithm to decode lattices constructed by Construction  $\mathbf{D}$ . Therefore properly selected lattices, such as those based on low-density generator matrix (LDGM) codes, can be decoded efficiently. The paper begins in the next section with a brief discussion about lattice. Section three introduces the generalized version of min-sum algorithm to decode lattices constructed by Construction  $\mathbf{D}$ . The decoding complexity and its bounds for the new generalization of min-sum algorithm are discussed in the forth section. The final section is dedicated to the paper's conclusions.

## 2. Preliminaries

Let  $\mathbb{R}^m$  be the  $m$ -dimensional real vector space with the standard product  $\langle \cdot, \cdot \rangle$  and Euclidean norm  $\| \mathbf{x} \| = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$ . A lattice  $\Lambda$  is a discrete additive subgroup of  $\mathbb{R}^m$ . An  $n$ -dimensional lattice is generated by the integer combinations of a set of  $n$  linearly independent vectors [3]. Any subgroup of a lattice  $\Lambda$  is called sublattice of  $\Lambda$  and a lattice is called *orthogonal* if it has a basis with mutually orthogonal vectors. The set  $\Lambda^*$  of all vectors in the real span of  $\Lambda$  ( $\text{span}(\Lambda)$ ), whose the standard inner product with all

elements of  $\Lambda$  has an integer value, is an  $n$ -dimensional lattice called the *dual* of  $\Lambda$ . Let us assume that an  $n$ -dimensional lattice  $\Lambda$  has an  $n$ -dimensional orthogonal sublattice  $\Lambda'$ . If  $\Lambda'$  has a set of basis vectors along the orthogonal subspaces  $S = \{W_i\}_{i=1}^n$ , the projection onto the vector space  $W_i$  defined as  $P_{W_i}$  and the cross section  $\Lambda_{W_i}$  defined as  $\Lambda_{W_i} = \Lambda \cap W_i$ . The *label group* of the lattice is defined as  $G_i = P_{W_i}/\Lambda_{W_i}$ , which is used to label the cosets of  $\Lambda'$  in  $\Lambda$ . For any lattice-word  $\mathbf{x}$ , the label sequence is defined as  $g(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_n(\mathbf{x}))$ , where  $g_i(\mathbf{x}) = P_{W_i} + \Lambda_{W_i}$ . The set of all possible label sequences,  $L = g(\Lambda) = \{g(\mathbf{x}) : \mathbf{x} \in \Lambda\}$ , is the label code of  $\Lambda$ . This set is an Abelian group block code over the lattice-alphabet sequence space,  $\mathbf{G} = G_1 \times \dots \times G_n$ . Let  $|G_i| = g_i$  and  $\mathbf{v}_i$  be the generator vector of  $\Lambda_{W_i}$ , *i.e.*,  $\Lambda_{W_i} = \mathbb{Z}\mathbf{v}_i$ . Each element of  $G_i$  can be rewritten in the form of  $\Lambda_{W_i} + j \det(P_{W_i})\mathbf{v}_i/|\mathbf{v}_i| (j = 1, \dots, g_i - 1)$ . Then the map

$$\Lambda_{W_i} + j \det(P_{W_i}) \frac{\mathbf{v}_i}{|\mathbf{v}_i|} \longrightarrow j \quad (1)$$

is an isomorphism between  $G_i$  and  $\mathbb{Z}_{g_i}$  [6], thus every element of the label group  $G_i$  can be written as  $(\mathbb{Z} + a_j)\mathbf{v}_i$ , where  $a_j = j \det(P_{W_i})/\det(\Lambda_{W_i})$ .

There exist another efficient method for lattice representation introduced by Tanner [4]. Lattices constructed by Construction **D** have a square generator matrix. If  $\mathbf{B}$  is a generator matrix for  $\Lambda$ , then  $\mathbf{B}^* = (\mathbf{B}^{-1})^{tr}$  is a generator matrix for  $\Lambda^*$  (parity-check matrix of  $\Lambda$ ) [2]. This can be applied to construct the Tanner graph for the lattice [3]. To construct the Tanner graph for a lattice, The Tanner graph construction of linear codes is applied to the parity check matrix of the lattice. If  $s_i$  denotes the  $i$ th column and  $ch_j$  denotes the  $j$ th row of  $\mathbf{B}^*$  respectively, then the Tanner graph of the lattice has the edge  $(s_i, ch_j)$  if and only if the lattice-word component (symbol node)

$s_i$  is contained in (or checked by) the parity-check sum (check node)  $ch_j$ . As a result  $\mathbf{x} \in \mathbb{Z}^n$  belongs to  $\Lambda$  if and only if  $\mathbf{B}^* \mathbf{x}^T \in \mathbb{Z}$ .

**Example 2.1** Consider the following  $\mathbf{B}$  and  $\mathbf{B}^*$  as the generator matrix and the parity check matrix of a 7-dimensional lattice constructed by Construction  $\mathbf{D}$ .

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}, \quad \mathbf{B}^* = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}$$

The corresponding Tanner graph for this example is illustrated in Fig. 1. The black circles denote lattice-word components and the white rectangles represent parity-check sums.

### 3. Generalized min-sum algorithm

In this section, we explain how generalized min-sum algorithm for lattices constructed by Construction  $\mathbf{D}'$  can be changed for decoding lattices constructed by Construction  $\mathbf{D}$ .

#### 3.1. Lattice decoding using min-sum algorithm

Given a vector  $\mathbf{y} \in \mathbb{R}^n$ , the lattice decoding problem is to find a lattice vector  $\mathbf{x}$ , such that  $\|\mathbf{y} - \mathbf{x}\|$  is minimized. Let  $\mathbf{y} = \sum_{i=1}^n \bar{y}_i \mathbf{v}_i$ , where

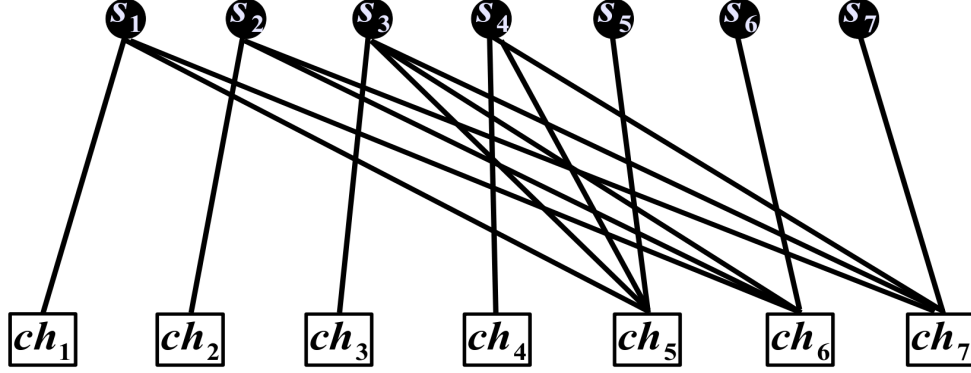


Figure 1: The corresponding Tanner graph for the 7-dimensional lattice discussed in example 2.1. The black circles and the white rectangles represent symbol nodes and check nodes respectively.  $s_i$  denotes the  $i$ th column and  $ch_j$  denotes the  $j$ th row of parity-check matrix  $\mathbf{B}^*$ .

$\bar{y}_i = \langle \mathbf{y}, \mathbf{v}_i \rangle / \langle \mathbf{v}_i, \mathbf{v}_i \rangle$ . The goal of the decoding algorithm is to search for the closest lattice point to the received-word. By definition:

$$x_j^{(i)} := a_j + \lceil \bar{y}_i - a_j \rceil, \quad i = 1, \dots, n \quad \text{and} \quad j = 0, \dots, |G_i| - 1 \quad (2)$$

where  $\lceil u \rceil$  is the closest integer to  $u$ ,  $a_j = j \det(P_{W_i}) / \det(\Lambda_{W_i})$  and  $x_j^{(i)}$  denotes the closest point in  $j$ th coset of  $G_i$  at the  $i$ th coordinate of the lattice-alphabet sequence space. Then  $x_j^{(i)} \mathbf{v}_i$  is the closest vector of  $G_i$  to  $\bar{y}_i \mathbf{v}_i$ . The set  $\mathbf{x}^{(i)} := \{x_j^{(i)} : j = 0, \dots, g_i - 1\}$  has  $g_i$  candidates of the  $i$ th coordinate of the lattice-alphabet sequence space for every component of the received-word. Define the weight as the squared distance between the elements of  $(\mathbb{Z} + a_j) \mathbf{v}_i$  and  $\bar{y}_i \mathbf{v}_i$ :

$$\omega_{y_i}(j) := (x_j^{(i)} - \bar{y}_i)^2 \|\mathbf{v}_i\|^2. \quad (3)$$

Considering the alphabet sequence space  $\mathbb{Z}_{g_1} \times \dots \times \mathbb{Z}_{g_n}$ , we rename  $j$  to  $c_i$  where  $c_i \in \mathbb{Z}_{g_i}$ . The weight of any valid codeword  $\mathbf{c} = (c_1, \dots, c_n) \in L$

correspond to lattice-word  $\mathbf{x} = (x_{c_1}^{(1)}, \dots, x_{c_n}^{(n)}) \in \Lambda$ , is defined as

$$\omega_{\mathbf{y}}(\mathbf{c}) = \sum_{i=1}^n \omega_{y_i}(c_i). \quad (4)$$

Now the problem is to find the minimum weight,  $\min(\omega_{\mathbf{y}}(\mathbf{c}))$ .

### 3.2. Min-Sum algorithm for Construction $\mathbf{D}$ lattices

This algorithm includes initialization, hard decision, symbol node operation and check node operation.

*Lemma 3.1:* For any received vector  $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$  we have

$$\omega_{y_i}(j) = \left( \det(\Lambda_{W_i}) \left( \left( \frac{y_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right) - \left\lceil \frac{y_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right\rceil \right) \right)^2, \quad (5)$$

where  $i = 1, \dots, n$  and  $j = 0, \dots, g_i - 1$ .

*Proof.* Proof is given in [6]. □

- 1) **Initialization:** Let  $\mathbf{y} = \sum_{i=1}^n \bar{y}_i \mathbf{v}_i = \sum_{i=1}^n y_i \mathbf{e}_i \in \mathbb{R}^n$  be the received vector. An initial weight is assigned to each node:

$$\omega_{y_i} = (\omega_{y_i}(0), \dots, \omega_{y_i}(g_i - 1)) \quad (6)$$

- 2) **Iteration:** In the iteration step all weights are alternatively updated to find the lattice-word.

2.1) **Symbol node operation:** The intermediate weight, which denotes the symbol-to-check outgoing weight, computed as follows:

$$\omega_{y_i, ch}(k) := \omega_{y_i}(k) + \sum_{\substack{ch' \in Q \\ y_i \in ch' \\ ch' \neq ch}} \omega_{ch', y_i}(k), \quad 0 \leq k \leq g_i - 1 \quad (7)$$

2.2) **Check node operation:** The intermediate weight, which denotes the check-to-symbol outgoing weight, computed as follows:

$$\omega_{ch,y_i}(k) := \min_{\substack{y_i \in Q_{ch}^{(i)} \\ y_i = x_k}} \sum_{\substack{y_{i'} \in ch \\ y_{i'} \neq y_i}} \omega_{y_{i'},ch}(k'), \quad 0 \leq k' \leq g_{i'} - 1 \quad (8)$$

where  $Q$  denotes the set of check equations and  $Q_{ch}$  denotes the set of all valid configurations that satisfy the  $ch$ th check equation.

3) **Termination:** For every symbol node, all incoming messages add to its initial weight to obtain final weight as follows:

$$F\omega_{y_i}(k) := \omega_{y_i}(k) + \sum_{\substack{ch' \in Q \\ y_i \in ch'}} \omega_{ch',y_i}(k). \quad (9)$$

The goal of the min-sum algorithm is to find a vector  $\mathbf{x} = (x_{c_1}^{(1)}, \dots, x_{c_n}^{(n)})$ , which  $x_k^{(i)} \in \{x_0^{(i)}, \dots, x_{g_i-1}^{(i)}\}$  and the index  $k$ , obtained as follows:

$$k = \arg(\min_{0 \leq k \leq g_i-1} F\omega_{y_i}(k)). \quad (10)$$

The iteration will stop when the selected vector  $\mathbf{x}$  is a lattice-word, *i.e.*,  $\mathbf{x}$  satisfies all parity check equations or reaches the maximum number of iteration.

#### 4. Decoding complexity

In each *Termination* step for every lattice-word component (symbol node) with label group size  $g_i$ , the number of comparisons is  $g_i - 1$ . Therefore the

total number of comparisons is:

$$\left( \sum_{i=1}^n g_i \right) - n. \quad (11)$$

In each *Iteration* step, the number of operations for each symbol node and check node computed separately:

a) For each symbol node and for each edge, the number of summation is  $g_i(d_{y_i} - 1)$ , where  $d_{y_i}$  denotes the number of edges for each symbol node. Thus the total number of summation would be:

$$\sum_{i=1}^n g_i d_{y_i} (d_{y_i} - 1). \quad (12)$$

b) For each check node, at most,  $g_{i_1} \times \dots \times g_{i_{d_{ch_i}}}$  comparisons should be made. For each outgoing message,  $\omega_{ch_i, y_k}(j)$ , and for each  $j$ , the number of summation is  $d_{ch_i} - 1$ , where  $d_{ch_i}$  denotes the number of edges of each check node. Since there are  $d_{ch_i}$  edges, the number of summation will not exceed  $d_{ch_i}(d_{ch_i} - 1)(g_{i_1} \times \dots \times g_{i_{d_{ch_i}}})$  summations for each check node. Then the total number of operations in all check nodes in each iteration is at most:

$$\sum_{i=1}^n d_{ch_i} (d_{ch_i} - 1) (g_{i_1} \times \dots \times g_{i_{d_{ch_i}}}). \quad (13)$$

Eqs. (11), (12) and (13) show the dependency of the decoding complexity on the size of label groups. In each iteration the total number of operations is:

$$\left( \sum_{i=1}^n g_i + g_i d_{y_i} (d_{y_i} - 1) + d_{ch_i} (d_{ch_i} - 1) (g_{i_1} \times \dots \times g_{i_{d_{ch_i}}}) \right) - n. \quad (14)$$

The next Corollary follows from counting the number of operations and inequality  $g_i \geq (\gamma(\Lambda)\gamma(\Lambda^*))^{1/2}$ , where  $\gamma(\Lambda)$ , denotes the coding gain of the



lattice [1].

*Corollary 4.1: (Bounds on decoding complexity)* Let  $\Lambda^*$  be the dual of  $\Lambda$  and  $\gamma = (\gamma(\Lambda)\gamma(\Lambda^*))^{1/2}$ . Also assume that  $\Lambda$  has a Tanner graph with  $n$  symbol nodes and  $n$  check nodes for which  $g_i \leq g$ ,  $d_y \leq d_{y_i} \leq d_y^{max}$  and  $d_{ch} \leq d_{ch_i} \leq d_{ch}^{max}$  ( $i = 1, \dots, n$ ). The upper bound of decoding complexity per iteration is:

$$n \left( g d_y^{max} (d_y^{max} - 1) + g^{d_{ch}^{max}} d_{ch}^{max} (d_{ch}^{max} - 1) + g - 1 \right) \quad (15)$$

and the lower bound per iteration is:

$$n \left( \gamma d_y (d_y - 1) + \gamma^{d_{ch}} d_{ch} (d_{ch} - 1) + \gamma - 1 \right). \quad (16)$$

The proof is a direct consequence of Eq. (14) and the fact that  $g_i \geq \gamma$  which has been shown in [1].

This corollary shows that the decoding complexity per iteration, grows linearly with the lattice dimension,  $n$ , but has power law dependence on the check nodes degree.

## 5. Conclusion

In this work the min-sum algorithm is generalized to decode lattices constructed by Construction **D**. It is shown that the upper and lower bounds of decoding complexity depends on lattice parameters like label group sizes, coding gain, check nodes and symbol nodes degree of Tanner graph. It is also shown that the decoding complexity grows linearly with the lattice dimension,  $n$ , but has the power law dependence on the check nodes degree. The

analysis of decoding complexity confirms the usefulness of LDGM codes for the lattice construction. It is worth mentioning that the presented decoding algorithm can be used to decode other constructions of lattices from linear codes.

## References

- [1] A.H. Banihashemi, F.R. Kschinschang, Tanner graphs for group codes and lattices: construction and complexity, *IEEE Transactions on Information Theory* 47 (2001) 824-882.
- [2] E.S. Barnes, N.J.A. Sloane, New lattice packings of spheres, *Canadian Journal of Mathematics* 35 (1983) 117-130.
- [3] J.H. Conway, N.J.A. Sloane, *Sphere Packing, Lattices and Groups* (3rd ed.), Springer-Verlag, New York, 1999.
- [4] G.D. Forney Jr., Density/length profile and trellis complexity of lattices, *IEEE Transactions on Information Theory* 40 (6) (1994) 1753-1773.
- [5] R. Kannan, Minkowski's convex body theorem and integer programming, *Mathematics of Optimizations Research* 12 (3) (1987) 415-440.
- [6] M.R. Sadeghi, A.H. Banihashemi, D. Panario, Low Density Parity Check Lattices: Construction and Decoding Analysis, *IEEE Transactions on Information Theory* 52 (2006) 4481-4495.
- [7] R.M. Tanner, A recursive approach to low complexity codes, *IEEE Transactions on Information Theory*, 27 (1981) 533-547.